PL1 - Travaux pratiques - Séance 2

Expressions et Types

Avant de commencer cette séance :

1. placez-vous dans votre répertoire ***PL1*** : `cd PL1`
2. créez un répertoire ***TP2*** : `mkdir TP2`
3. placez-vous dans le répertoire ***TP2*** : `cd TP2`
4. récupérez les fichiers nécessaires à ce TP :
   `cp ~mounlaur/CCI_PL1/TP2.tar.gz .`
5. dé-compressez et dé-archiver ce fichier :
   `gunzip TP2.tar.gz ; tar -xvf TP2.tar`

Exercice 1 - Racines d'un polynôme du second degré [obligatoire]

On se rappelle que pour résoudre dans $\mathcal{R}$ une équation E du 2^{nd} degré de la forme $ax^2 + bx + c$ on peut procéder de la manière suivante :

1. calculer le discrimant $\Delta = b^2 - 4ac$
2. selon Δ
   — cas $\Delta < 0$: E n'admet pas de solution dans $\mathcal{R}$
   — cas $\Delta = 0$: E n'admet une seule solution $x_0 = -b/(2a)$
   — cas $\Delta > 0$: E n'admet deux solutions $x_1 = (-b + \sqrt{\Delta})/(2a)$ et $x_2 = (-b - \sqrt{\Delta})/(2a)$

Q1. Ecrivez dans un fichier `equation2.c` un programme C qui lit au clavier 3 réels a, b et c et affiche la (ou les) solution(s) de l'équation E correspondante.

Q2. Compiler votre programme (avec le compilateur gcc) et exécutez-le sur des entrées permettant de produire les 3 résultats possibles (pas de solution, une solution unique, deux solutions distinctes)

Q3 (*) Pour vérifier si votre programme est "correct" vous pouvez le compléter en lui ajoutant une étape de vérification qui consiste à calculer la valeur de E avec la (ou les) solution(s) obtenue(s) et à afficher un message d'erreur si le résultat n'est pas égal à 0. Attention, il faudra sans doute prendre en compte des problèmes d'arrondis ...

Exercice 2 - A propos de dates [obligatoire]

Le fichier ***date.c*** contient la définition d'un type `Date` permettant de représenter des dates sous la forme d'un produit de type `Jour` × `Mois` × `Annee`. Lisez ce fichier.

Q1. Complétez ce fichier pour écrire un programme C permettant de :

1. Lire une date au clavier (en lisant successivement un `Jour`, un `Mois` et une `Annee`).
2. Afficher cette date à l'écran

3. Afficher un message indiquant si cette date est "correcte " ou non. On considérera qu'une date est correcte si elle remplit les conditions suivantes :
   — le jour est un entier entre 1 et 28 (ou 29, ou 30, ou 31, selon le mois et l'année considérés) ;
   — les années bissextiles (qui ont 29 jours en février) sont celles qui sont soit divisibles par 4 mais pas par 100, soit divisibles par 400.

Indications :
— pour lire une date au clavier vous pouvez saisir un entier pour le mois (1 pour janvier, 2 pour février, etc.) [1]
— pour afficher une date à l'écran vous pouvez utilisez la fonction `afficheDate` fournie ;

Q2. Complétez maintenant ce programme pour qu'il lise deux dates au clavier, et, si elles sont correctes, les affiche par ordre chronologique.

Exercice 3 - Heures, minutes, et secondes [optionnel]

Q1. Ecrivez un programme C qui lit une valeur entière positive au clavier supposée représenter une durée en secondes, convertit cette durée en heure, minutes, secondes, et affiche le résultat.
Par exemple, si la valeur lue est 5132 le programme devra afficher `1h 25mn 32s`.

Exercice 5 - Débordements arithmétiques [conseillé]

[Cet exercice sera corrigé en cours]

On rappelle qu'en langage C :
— une valeur entière non signée codée sur n bits est un élément de l'intervalle $[0, 2^n - 1]$;
— les opérations arithmétiques sur des entiers non signés se font modulo 2^n.
Lisez le contenu du fichier ***debordechar.c***, et vérifiez que vous comprenez ce qui va se passer lors de l'exécution. Compilez-le sous le nom ***debordechar***, et exécutez-le.

En vous aidant des résultats affichés, répondez aux questions suivantes : Quelle est la taille en en bits d'une variable de type `unsigned char` ? Quelle est cette taille en octet ? Quel est le plus grand entier représentable avec le type `unsigned char` ?

Créez une copie de ***debordechar.c*** sous le nom ***debordeshort.c***. Modifiez ce fichier afin qu'il serve à tester des variables de type `unsigned short int` (au lieu de `unsigned char`), et répondez aux mêmes questions que ci-dessus, pour le type `unsigned short int`.

Créez une copie de ***debordeshort.c*** sous le nom ***debordeint.c***. Modifiez ce fichier afin qu'il serve à tester des variables de type `unsigned int` (au lieu de `unsigned short int`), et répondez aux mêmes questions que ci-dessus, pour le type `unsigned int`.

Par quel facteur (environ) est multiplié le temps d'exécution entre **debordechar** et **debordeint** ?
Si l'on définit un type d'entiers non signé représentés sur 8 octets, et qu'on effectue la même expérience, combien de temps (environ) faudra-t-il attendre pour voir s'afficher le résultat ? Vous pouvez pour cela estimer le facteur multiplicatif entre le nombre d'itérations effectuées par **debordeint** et celui effectuées par **debordechar** ...

1. en vous rappelant que les types énumérées sont représentés par des entiers commençant à 0 ...

Exercice 6 - Erreurs de type et conversions de type [optionnel]

Le fichier ***erreur_type.c*** contient un programme C qui comporte plusieurs erreurs de type.

Q1. Compilez ce programme pour trouver ces erreurs.

Q2. Corrigez ces erreurs, et vérifiez que le nouveau programme se compile maintenant correctement (sans erreurs).

Q3. Exécutez ce programme et examinez l'effet des conversions de types implicites effectuées à l'exécution.